0/9 Questions Answered

Exam

Q1 Forward Modeling -- Wave propagation
20 Points

Q1.1 Finite Difference
5 Points

Consider the one-dimensional acoustic wave equation:

$$\frac{\partial^2 u}{\partial t^2} = c^2 \frac{\partial^2 u}{\partial x^2}, \quad x \in [0, L], \quad t \in [0, T]$$

where $u(x, t)$ is the displacement field, and c is the wave speed.

Part A: Forward Problem using Finite Difference Method

Solve the wave equation using the finite difference method with the following specifications:

Domain and Parameters:

- Spatial domain: $x \in [0, 1]$ (i.e., $L = 1$)
- Time domain: $t \in [0, 2]$ (i.e., $T = 2$)
- Wave speed: $c = 1.0$
- Use appropriate spatial and temporal discretization (verify CFL stability)

Initial Conditions:

$$u(x, 0) = \sin(\pi x) + 0.5 \sin(3\pi x)$$

$$\frac{\partial u}{\partial t}(x, 0) = 0$$

Boundary Conditions:

$$u(0, t) = 0$$

$$u(L, t) = 0$$

Implement a second-order accurate finite difference scheme in both space and time.

Check for time-step stability
Space-time contour plot of $u(x, t)$
Solution snapshots at times $t = 0, 0.5, 1.0, 1.5, 2.0$ s

Please select file(s) Select file(s)

Save Answer

Q1.2 PINNs Forward
15 Points

Solve the same wave equation using PINNs with the same parameters and conditions as Part A using soft boundary constraints.

Use collocation points sampled in the interior of the domain, along with points at the boundaries and initial time. Compute derivatives using automatic differentiation.

Deliverables:

i) Training loss curves (total and individual components)

ii) Space-time contour plot of the PINN solution

iii) Solution snapshots at $t = 0, 0.5, 1.0, 1.5, 2.0$ s

iv) Comparison with FD solution (absolute error plot) at 2.0 s.

Please select file(s) Select file(s)

Save Answer

Q2 Inverse Analysis
30 Points

Q2.1 Inverse with AD
10 Points

Given noisy observations of the wave field $u(x, t)$ at specific sensor locations, recover the unknown wave speed c.

Forward Model: Use the finite difference scheme from Part A implemented in a differentiable framework (JAX, PyTorch, or TensorFlow) for a target linear wave velocity of 0.9 -- 1 m/s increasing linearly from 0 m to 1 m.

Use automatic differentiation to compute the gradient $\frac{d\mathcal{J}}{dc}$ and apply gradient-based optimization. Start from an initial guess c_0 varying linearly from 0.85 -- 1.0 with 10,000 sampling points in space.

Deliverables:

i) Convergence of the objective function $\mathcal{J}$ vs. iteration

ii) Evolution of estimated wave speed c and compare with true value

iii) Final simulated solution vs. observations at each sensor location at $T = 2.0$ s

iv) Explain the difference between forward and reverse mode AD and explain when will you use which approach?

Please select file(s) Select file(s)

Save Answer

Q2.2 PINNs Inverse
20 Points

Given the same observations as above, use PINNs to simultaneously learn the solution $u(x, t)$ and discover the unknown wave speed c in the governing equation.

Analysis: Discuss the accuracy of the discovered wave speed, effect of noise on parameter estimation use a 2% and 5% Gaussian noise, and comparison with the automatic differentiation approach.

Please select file(s) Select file(s)

Save Answer

Q3 Beam on Elastic foundation

33 Points

Consider a beam resting on an elastic foundation described by the Winkler model:

$$EI\frac{d^4w}{dx^4} + kw(x) = p(x), \quad x \in [0, L]$$

where:

- $w(x)$ is the transverse deflection of the beam
- EI is the flexural rigidity (bending stiffness)
- k is the foundation modulus (stiffness per unit length)
- $p(x)$ is the distributed load
- L is the length of the beam

Parameters:

- Beam length: $L = 10$ m
- Flexural rigidity: $EI = 5.0 \times 10^6$ N·m^2
- Foundation modulus: $k = 1.0 \times 10^5$ N/m^2

Boundary Conditions (Simply Supported):

$$w(0) = 0, \quad w(L) = 0 \quad \text{(zero deflection)}$$

$$\frac{d^2w}{dx^2}(0) = 0, \quad \frac{d^2w}{dx^2}(L) = 0 \quad \text{(zero moment)}$$

Q3.1 Finite Difference
5 Points

Implement a finite difference solver for the Winkler beam equation using the standard five-point stencil for the fourth derivative:

$$\left.\frac{d^4 w}{dx^4}\right|_i \approx \frac{w_{i-2} - 4w_{i-1} + 6w_i - 4w_{i+1} + w_{i+2}}{\Delta x^4}$$

This solver will serve as both the reference solution and the data generator for training the DeepONet.

Test Load Case: Use a sinusoidal load:

$$p(x) = q_0 \sin\left(\frac{\pi x}{L}\right), \quad q_0 = 1000\ \mathrm{N/m}$$

Deliverables:

i) Deflection profile $w(x)$ for the test load

ii) Verification against analytical solution: $w(x) = \frac{q_0 \sin(\pi x/L)}{EI(\pi/L)^4 + k}$

iii) Report computational time per solve

Please select file(s) Select file(s)

Save Answer

Q3.2 DeepONet
25 Points

Implement a Deep Operator Network (DeepONet) to learn the mapping from distributed loads $p(x)$ to beam deflections $w(x)$.

Train the DeepONet to minimize the mean squared error between predicted and true deflections:

$$\mathcal{L} = \frac{1}{N_{\text{train}}} \sum_{i=1}^{N_{\text{train}}} \|w_{\text{DeepONet}}(p_i) - w_{\text{FD}}(p_i)\|^2$$

Deliverables:

i) Training loss curve (log scale)

ii) Test set MSE (both normalized and physical units)

iii) Example predictions: plot predicted vs. true deflection for 3-5 test cases

iv) Report total training time (wall-clock)

v) Compare the DeepONet predictions against FD solutions for 5 sample load displacements.

Please select file(s) Select file(s)

Save Answer

Q3.3 DeepONet
3 Points

List 3 main limitations of the DeepONet architecture

Save Answer

Q4 MLP
2 Points

You are training an MLP for a binary classification task and it is suffering from severe overfitting. Describe three distinct techniques you could use to mitigate this issue and briefly explain the mechanism behind each

Save Answer

Q5 MLP Activation function
15 Points

Neural Network Initialization and Activation Functions

Exam Question: Investigating MLP Behavior for PDEs

Problem Statement

You will train a Multi-Layer Perceptron (MLP) to solve the 1D Poisson equation:

$$-\frac{d^2u}{dx^2} = f(x), \quad x \in [0, 1]$$

with boundary conditions: $u(0) = 0$, $u(1) = 0$

Using source term $f(x) = \pi^2 \sin(\pi x)$, which has analytical solution $u(x) = \sin(\pi x)$.

Network Architecture

- **Input**: $x \in [0, 1]$
- **Hidden layer**: One hidden layer with 32 neurons.
- **Output**: $u(x)$ (scalar)
- **Loss function**: MSE between predicted $u(x)$ and analytical solution

Questions

Part A: Initial Setup

Implement an MLP with the following specifications:
1. Initialize all weights uniformly in the range $[-1, 1]$
2. Use ReLU activation functions
3. Train for 2000 epochs using Adam optimizer

After training:

- Plot the predicted solution vs. analytical solution
- Report the final MSE loss
- **Track and report**: For each hidden layer, count how many neurons have maximum activation = 0 across all training samples

Part B: Analysis

Answer the following:

1. What percentage of neurons in each layer produce zero output for all inputs? Does this percentage change during training?
2. Explain why this phenomenon occurs. In your explanation, address:

   - Why does uniform initialization in $[-1, 1]$ cause this behavior with ReLU?
   - What is the mathematical relationship between neuron input and ReLU output?
   - Why can't these neurons recover during training? (Hint: consider gradients)

3. How does this affect:
   - The effective capacity of your network?
   - Training convergence and final accuracy?
   - Your ability to approximate the true solution?

Part C: Solutions

Test the following alternatives and compare results:

1. **He initialization** (Kaiming): $\mathcal{N}(0, \sqrt{2/n_{in}})$ with ReLU
2. **Leaky ReLU** ($\alpha = 0.1$) with uniform $[-1, 1]$ initialization
3. **GELU** activation with uniform $[-1, 1]$ initialization

For each configuration:

- Report the percentage of inactive neurons
- Plot the solution approximation
- Report final MSE

Explain: Why does each alternative mitigate (or not mitigate) the problem you identified in Part B?

Please select file(s) Select file(s)

Save Answer

Save All Answers

Submit & View Submission